



ALLAN THRAEN |

🕒 16 years ago |



PIMP UP YOUR SEARCH (PART I)



Let's face it, the EPiServer CMS built-in search isn't the best in the world. Neither is it intended to be - there are a lot of 3rd party search engines available for EPiServer that are great at what they are doing - and in general I highly recommend using one of their products if search is essential to your web site. However, for those who feel the built-in search is *almost adequate* for their needs, I've decided to share a couple of tips on how to pimp up the built-in search (and some of the tips might even apply to other search engines as well).

This, first post in the series, will demonstrate how you can add 3 cool (and popular) features to your EPiServer CMS search, just by changing a bit on the result-page and adding some pages in Edit-mode.

The features we will add are these:

1. **Sponsored Text / Banner / Link.** *For a given search query, a piece of custom xhtml will be displayed on top of the search results. Ideal for marketing campaigns, etc. Example: A visitor searches for "Couches" and you just happen to have a specific couch on sale that week - then put a picture and a link to it directly on top of the search results. Or if a visitor searches for "newsletter", then why not allow him to sign up directly on the result page?*
2. **Synonyms / Did-You-Mean feature.** *For a given search query, alternative queries are suggested. Can be used to prevent common misspellings, map from competitive product-names, etc.*
3. **Direct Links.** *For a given search query, send the user directly to a specified page without showing the other results. Use with care. This is handy in the rare cases when there only is one (!) relevant result for a given query. For instance if a visitor searches for "Contact" or "Address" or "Phone number". In those cases you probably want to direct them directly to the "About Us / Contact Us" page.*

HOW DO WE DO IT?

First of all, we need a place for the editors to set up how they want these features configured. But that part is really easy - since we want this to interact with the search, why not put it directly in the tree structure as pages? By putting all the search actions as pages in a structure below the root node (but outside the actual active web site) we can use the search to find the actions - and just display them in a different way than it would display normal results.

So, I define a *SearchAction* Page type in Admin mode (doesn't matter what template it points at - as long as the path is filled out) and I add the properties I want the editors to be able to set. They need to be able to set which keywords each action should match. The property for that should be a string property, and it should be marked as Searchable. There is already a build-in property which is a searchable string, and that's the PageName, but I add an extra just in case they want to fill in more words than the name can contain. Asides from that I add a property to hold the sponsored text, a property to hold the synonym and a Page property to hold the link to a page for direct links.

We go to Edit-mode and create a few SearchActions. In the PageName / Search Words fields we'll put the words to match on. If we want an Action to be a sponsored Link / banner, we fill out that field. Alternative we can set a direct link or a Did-You-Mean suggestion. When we're done it's just a matter of choosing Save & Publish, the built-in search will automatically index the searchable fields.

In order to structure it in a way where it will be easy to work with, I created an Empty node called "[SearchActions]" directly below the root, not below the start page of the web site. That way it will not be searched in the usual web site search that searches below the Start page.

I also marked all the actions as not visible in Menu, just to avoid them showing up places they shouldn't.

On my sandbox site, allan.epidesk.com I created a Sponsored text for the search word "Allan", a "Did You Mean" for various misspellings of EPiServer like 'epi', 'epserver', etc. and a direct link / redirect for 'contact', 'address', 'email'. Feel free to try it out!

FIXING THE RESULT PAGE

All it takes code-wise to implement these features are simply a few modifications to the result page. In my case I've just modified the Search.aspx from the Public package. First you want to add an extra EPiServer:SearchDataSource control that just searches in the "[SearchActions]" branch. I choose to add the definition below the original SearchDataSource.

```
<EPiServer:SearchDataSource ID="ExtraSearchSource" runat="server"
PageLink="169" IncludeRootPage="true" OnlyWholeWords="true" EnableVisibleInMenu="false">
  <SelectParameters>
    <asp:ControlParameter runat="server" ControlID="SearchText" Name="SearchQuery" PropertyName="Text" />
  </SelectParameters>
</EPiServer:SearchDataSource>
```

In this example I've simply hardcoded the PageLink to the ID of the "[SearchActions]" root page, however you might want to make this depend on a property. Also note the "EnableVisibleInMenu=false" which is needed because all the actions are set to not being displayed in menus.

The only other modification needed was to repeater controls. The first displays the "Did You Mean" texts and redirects (using a few lines of code) direct links, the other displays any Sponsored Texts / banners.

```
>
<script runat="server">

    public string CreateDidYouMean(object o)
    {
        EPiServer.Core.PageData pd = (EPiServer.Core.PageData)o;
        if (pd["SearchLink"] != null){
            //We have a Direct link - redirect
            Response.Redirect(
                GetPage((EPiServer.Core.PageReference)pd["SearchLink"]).LinkURL, true);
        }
        if (!string.IsNullOrEmpty((string)pd["DidYouMean"])){
            //Create "Did You Mean" text
            return "<br/>Did you mean <a href=\"" +
                CurrentPage.LinkURL +
                "&quicksearchquery=\"" +
                Server.UrlEncode(pd["DidYouMean"].ToString()) +
                "\">" + pd["DidYouMean"].ToString() + "</a>";
        }
        return string.Empty;
    }

</script>
<asp:Repeater ID="Repeater1" runat="server" DataSourceID="ExtraSearchSource">
    <ItemTemplate>
        <# CreateDidYouMean(Container.DataItem) %>
    </ItemTemplate>
</asp:Repeater><br />
<asp:Repeater ID="Repeater2" runat="server" DataSourceID="ExtraSearchSource">
    <ItemTemplate>
        <EPiServer:Property runat="server" PropertyName="Banner" />
    </ItemTemplate>
</asp:Repeater>
```

Notice how the first repeater uses a Method to render the Did-You-Mean feature, which at the same time is able to redirect if it's needed.

It's a quick and dirty solution, but it works and can be very efficient in allowing your editors to customize the search.